A Threshold Based Multi-Label Classification

Durga Prasad Muni, Bintu G. Vasudevan and Rajesh Balakrishnan  
Infosys Labs  
Infosys Limited  
Bangalore, India  
e-mail: {DurgaPrasad_Muni, Bintu_Vasudevan, RajeshB}@infosys.com

Abstract—In classification problems, a pattern may belong to one or multiple categories. It is essential to deal multi-label classification accurately and efficiently. Threshold strategies can be used for multi-label classification. We propose four schemes to compute threshold for a threshold based multi-label classification. We validate our method using multi-label text data and multi-label image data.

Keywords-multi-label, threshold, classification, data mining

I. INTRODUCTION

In a typical pattern classification problem, a single label/class is assigned to a given pattern from a pre-defined set of labels or classes. However, in many classification problems such as text categorization, image classification, music classification and medical diagnosis, a pattern (text/image/music/patient) may belong to more than one class or category simultaneously. For example, a news article describing a person who is both a player and a politician can be classified into two categories - Sports and Politics. In multi-label classification, we assign one or multiple labels to a given pattern from a pre-defined set of labels or categories.

An overview of multi-label classification is given in [1]. The multi-label classification problem can be considered as a set of binary classification problems. For each category, this approach constructs a classifier by using data associated with this category as positive and all others as negative. For a given pattern, a probabilistic score indicative of the membership to the category is expected from the corresponding classifier. The threshold strategies can be applied to the set of scores to predict categories of a given pattern.

Yang [2] presented various threshold strategies and compares their effectiveness. The author grouped the threshold strategies into 3 groups: rank-based cut, proportion-based cut and score-based cut. Fan and Lin investigated selection of thresholds for score-based multi-label classification in [3]. The authors tuned the decision thresholds of the binary classifiers. Tang et. al. [4] proposed an approach called Metalabeler. It determined the relevant set of labels by learning the expected number of labels.

Sanden and Zhang [5] proposed a set of ensemble techniques specific to multi-label music genre classification. Wicker et. al. [6] proposed a multi-label classifier based on boolean matrix decomposition. Boolean matrix decomposition was used to extract latent labels representing useful boolean combinations of the original labels from the full label matrix. Cerri et. al. [7] used genetic algorithms (GA) for hierarchical multi-label classification. GA evolved the antecedents of classification rules. The set of evolved antecedents is selected to build the corresponding consequent of the rules. In [8], voting based learning classifier system was proposed for multi-label classification. In [9], a hypergraph spectral learning formulation was proposed for multi-label classification, where the hypergraph exploits correlation among class labels. To reduce computational cost, the authors proposed an approximate formulation that is equivalent to a least squares problem. To exploit dependencies between labels, Ghamravi and McCallum [10] proposed multi-label conditional random field classification models that parameterize label co-occurrences in multi-label classification. Wang et. al. [11] proposed random-walk model based multi-label classification system. It mapped the multi-label patterns to graphs, on which the random walk was applied. For a given unlabeled pattern, the system transformed the original multi-label problem to some single-label sub-problems. Harrihan et. al. [12] proposed a max-margin formulation for the multi-label classification problem. The authors assumed labels are correlated but does not incorporate pairwise label terms in the prediction function. They also developed efficient optimization algorithms that are orders of magnitude faster than existing (cutting plane) methods.

In this paper, we present four schemes to determine threshold for multi-label classification. In section II, we give a brief introduction to multi-label classification with respect to threshold approach. In section III, we present our approach of threshold based multi-label classification. In section IV, we provide the experimental results and in section V we conclude.

II. MULTI-LABEL CLASSIFICATION

Let $X = \{x_1, x_2, ..., x_N\}$ be the given set of patterns. $N$ is the size of the data set. Let $\{C_1, C_2, ..., C_L\}$ be the predefined categories. $L$ is the total number of categories. A pattern $x_i$ may belong to one or more categories. Let a set of
labels, \( S_i \), be associated to \( x_i \), where \( |S_i| \geq 1 \). The data set \( D = \{(x_i, S_i), \forall i \in \{1,2,\ldots,N\}\} \). Suppose \( x_i \in C_i \), \( x_k \in C_j \) and \( x_l \in C_d \). Then the set of labels \( S_2 = \{C_1, C_3, C_4\} \) is assigned to \( x_2 \).

The multi-label classification problem can be considered together as a set of binary classification problems. In this approach, for each class \( C_j \), a data set \( D_j \) is prepared from \( D \). In \( D_j \), all patterns \( \{x_i\} \in C_j \) are included with class label 1, i.e. the pairs \( \{x_i, 1\} \) are included to \( D_j \) and all patterns \( \{x_i\} \notin C_j \) are included with class label 0 (or -1). For class \( C_j \), a classifier \( f_j \) is learned using the data set \( D_j \). It is assumed that the classifier \( f_j \) provides a score \( f_j(x) \) for a pattern \( x \) indicating the class membership of \( x \) to class \( C_j \). This set of scores \( \{f_1, f_2, \ldots, f_j\} \) is used to determine the set of labels \( (S_j) \) of the given pattern \( x \). Threshold strategies can be applied on this set of scores to predict the class labels of \( x \). In [2], the threshold strategies are described into three categories:

- **Rank-based Cut (RCut):** It sorts scores for each pattern \( x \) and ranks the categories. Then the top \( k \) categories are assigned to \( x \). Usually, \( k \) is defined as the average length of labels in the training data [4]. If average length is in between 2 and 3 then \( k \) can be taken either 2 or 3.

- **Proportion-based Cut (PCut):** Here, for each category \( C_j \), the test patterns are sorted by the scores for \( C_j \) and the class \( C_j \) is assigned to the top \( k \) patterns. The \( k \) is defined based on the prior probability of \( C_j \) estimated on training data. Since, test data as a batch is needed, so in real-world applications it is rarely used.

- **Score-based local optimization (SCut):** It tunes the threshold for each category using a validation set. SCut optimizes the performance of the classifier on individual categories without guaranteeing a global optimum. This method is studied by Fan and Lin [3].

We have compared our approach with baseline approaches [4]. These approaches are:

- a) Vanilla SVM (SVM): It is one-vs-Rest SVM without any post-processing technique. At the time of prediction, all the labels with a positive score are selected.

- b) RCut: We have used RCut with \( k \) equal to average number of labels per pattern. The nearest integer of this real average number value is taken as either:
  
  i) \( k = \text{Average Length} \). This RCut is denoted as RCut (RCut conservative).
  
  ii) \( k = \text{Average Length} \). This RCut is denoted as RCut (RCut aggressive).

- c) SCut: SCut tuned based on Micro-F1 is denoted as SCut, and SCut tuned based on Macro-F1 is denoted as SCut.

### III. MULTI-LABEL CLASSIFICATION: OUR APPROACH

For a given pattern, we assign class \( C_j \) to the pattern if score \( f_j \geq \theta_{avg} \) a threshold value. The value of \( \theta_{avg} \) is computed using validation sets. We partition the training set \( D \) into a training set \( D_v \) and a validation set \( (D_{val}) \) using 5-fold cross validation. We design a set of \( L \) binary classifiers using \( D_v \) as mentioned in the previous section. We use support vector machine (SVM) as the classifier. We apply the classifiers on the validation set. For each instance \( x \) in the validation set, we obtain a set of probabilistic scores \( \{f_1, f_2, \ldots, f_j\} \) corresponding to categories \( \{C_1, C_2, \ldots, C_L\} \).

We find a value \( \theta \) for each instance \( x \) of validation set. Then we take the average value of \( \theta \) over all instances in the validation set as \( \theta \). After computing \( \theta \) for each validation set of 5-fold cross validation, we take the average of the five \( \theta \) values to obtain threshold value \( \theta_{avg} \).

We propose four approaches to compute \( \theta \):

1) **Approach1:** In the first approach, we sort the scores \( \{f_1, f_2, \ldots, f_j\} \) of validation instance \( x \) in descending order. If \( k \) is the total number of labels are associated to \( x \), then we take the \( k \)th score of the sorted scores of \( x \) as \( \theta \).

2) **Approach2:** In the second approach, \( \theta \) is the minimum of the \( k \) scores corresponding to the \( k \) labels that are associated to \( x \).

3) **Approach3:** In the third approach, if \( k \) is the total number of labels are associated to \( x \), then we consider \( (k+dy) \) score after sorting the scores in descending order. Also, we consider the minimum of \( k \) scores corresponding to the \( k \) labels associated to \( x \). The larger of these two scores is taken as \( \theta \). For RCV1 data, we have taken \( d = 2 \) and for scene data we have chosen the value of \( d \) as 1 (as number of total categories is small).

4) **Approach4:** In the fourth approach, if \( k \) is the total number of labels are associated to \( x \), then we consider the average of the \( k \)th score (after sorting) and the minimum of \( k \) scores corresponding to the \( k \) labels associated to \( x \) as \( \theta \).

After obtaining \( \theta_{avg} \) for above mentioned approaches, we train SVM with the complete training set \( D \) and then test the classifiers on the test data. While classifying test samples, we use corresponding \( \theta_{avg} \) of the above mentioned approaches to determine classes.

### IV. EXPERIMENTAL RESULTS

We used LIBSVM [13] tool to validate our approach. We implemented the 5-fold cross validation to optimize the parameters of SVM.

#### A. Data Sets

To validate our approach, we used the benchmark multi-label RCV1 five subset data sets [14] and multi-label scene classification data [15].

- **RCV1 (Reuters Corpus Volume I) Data:** It is a text data (news documents). It has five subsets. Each subset has 3000 data points for training and 3000 data points for test, with in total 103 categories (topics). Most instances are labeled with multiple labels. Two categories in the 5 training sets do not contain any instances. So, we have removed these two categories from test sets. After dropping these two classes, it contained 101 categories. The instances are represented by 47,236 features. The data sets are highly imbalanced.

- **Scene Data:** This is an image data. The task is to recognize which of six possible scenes available in the given set of images. These scenes are beach, sunset, field, fall foliage, mountain and urban. The data set contains 1211 pictures for training and 1196 pictures for testing. The
pictures are represented by 294 attributes. Few data points are labeled with multiple labels. That means, few images contain more than one scene.

In [16], it is mentioned that for text data, linear SVM performs better. So, we used linear SVM for learning the classifiers from RCV1 text data. However, for scene data, we used Radial Basis Function (RBF) SVM.

B. Evaluation Measures

We adopted the measures mentioned in [3] for multi-label classification performance measure. These measures are exact match ratio, Macro-F1 and Micro-F1. These measures are defined below. Let M be the total number of test patterns. Let \( y_i, \hat{y}_i \in \{0,1\}^L \) be the actual label set and the predicted label set for pattern \( x_i \) respectively.

1) Exact Match Ratio:

\[
\text{Exact Match Ratio} = \frac{1}{M} \sum_{i=1}^{M} I[y_i = \hat{y}_i]
\]

\( I \) is the indicator function. \( I[z] = 1 \), if \( z \) is true and 0 otherwise. Exact match ratio is the extension of the accuracy for traditional classification. It does not consider partial match between the actual labels and prediction labels. Macro-F1 and Micro-F1 consider partial matches.

2) Macro F1:

\[
\text{Macro-F1} = \frac{1}{L} \sum_{i=1}^{L} F_1^i
\]

\( F_1^i \) is the F1 measure of \( C_i \) category. F1 measure is the harmonic mean of precision and recall.

\[
F_1^i = \frac{2 \sum_{i=1}^{M} y_i^i \hat{y}_i^i}{\sum_{i=1}^{M} y_i^i + \sum_{i=1}^{M} \hat{y}_i^i}
\]

3) Micro F1:

\[
\text{Micro-F1} = \frac{2 \sum_{i=1}^{L} \sum_{i=1}^{M} y_i^i \hat{y}_i^i}{\sum_{i=1}^{L} \sum_{i=1}^{M} y_i^i + \sum_{i=1}^{L} \sum_{i=1}^{M} \hat{y}_i^i}
\]

C. Results

Tables I-V show the results using our approaches and using baseline methods for RCV1 subset data sets. For baseline methods, we have taken the result given in [4]. For all RCV1 subset data sets, our threshold based approaches are giving better Macro-F1 as compared to the baseline methods. Approach2 performs consistently well.
We obtained marginally better result for all 3 approaches to compute performance. We observed that binary approach for the multi-label classification and used these three measures to predict the class labels. We used 5-fold cross validation to compute threshold value for classification and to find the value of parameters of the SVM. We validated our approach using benchmark RCV1 multi-label data sets and multi-label Scene classification data set. We obtained consistently better Macro-F1 against baseline approaches. This indicates that our method could capture the rare categories well.

V. CONCLUSION

We proposed four approaches to compute threshold values for a threshold based multi-label classification. We used SVM to learn the classifiers from data. We decomposed the multi-label classification problem into a set of binary classification problems and then used threshold approach to predict the class labels. We used 5-fold cross validation to compute threshold value for classification and to find the value of parameters of the SVM. We validated our approach using benchmark RCV1 multi-label data sets and multi-label Scene classification data set. We obtained consistently better Macro-F1 against baseline approaches. This indicates that our method could capture the rare categories well.

REFERENCES

international Conference on Information and Knowledge Management (CIKM 11), October 2011, pp. 2197 – 2200.


